Process for new Device Services

This note outlines the process by which a contributed device service may be adopted into EdgeX. Consideration will generally be given to services which provide access to classes of devices previously unavailable in EdgeX, or which have substantial advantages over an existing implementation.

Initial proposal

In the first instance, the contributor should contact the working group chair to arrange a date on which consideration of the new service may be added to the WG agenda. At the arranged meeting, the contributor should give a short presentation on the new service and take questions. The working group will likely be interested in:

- What types of device are supported.
- What protocol features are supported / likely to be supported in future.
- What limitations does the existing implementation have.
- What hardware and software has it been tested against.

A brief demonstration of the device service in operation may be helpful.

If the WG approves, a repository in the edgexfoundry-holding project will be created, for the new service to be imported into.

Service Review

When the service has been imported into the -holding repository, and the contributor feels that it is ready to be reviewed, these steps are to be followed:

1. PR Created

- A new empty branch called "review" is created in the service's git repository in edgexfoundry-holding
    - This should be branched from the repo's initial commit
- A PR is created with "main" as the source branch, and "review" as the target branch
    - The PR should not be created by anyone who will be reviewing the service, as PR creators may not be assigned as reviewer on GitHub
- Ideally we create a PR template for reviews, but for now the PR should follow the general guidelines as described by the device services holding review process

2. Assign reviewers

Based on priorities set by the working group chair and/or TSC, the device services WG should ask for volunteers to review the service. A minimum of two reviewers is required. The reviewers should not be from the same group within the company that submitted the contribution. The WG chair should add both reviewers to the PR.

3. Due Date

The reviewers and the WG group chair should decide on a mutually agreeable due date for the initial reviews to be submitted, this should be recorded in the PR description (or comment) and the WG meeting notes. A baseline of two weeks from assignment should be a good starting point for discussion.

4. PR updates

If the reviewers submit "request changes" reviews, then the authors should address these and push new commits to address the feedback (i.e. no force pushes should be used). While it may not always be possible, ideally reviewers should re-review the PR within (3) days of an update to the PR.

5. WG Approval

Once a minimum of two "approved" reviews have been submitted, the WG shall vote on accepting the service. If this vote passes, then it should be recorded in the WG meeting minutes, and TSC should be notified. Meanwhile the PR should be squashed and merged.

6. TSC Approval

Once approved by the TSC, the LF should be notified that the repository can now be moved into the edgexfoundry organization on GitHub. At this point the review branch should be re-named to main.

Review Criteria

Other than general code quality, the review should consider:

- The service should implement the functionality described in the Device Services Requirements document. Services written in Go or C should achieve this by using the official SDK.
- The service should target either the current or development versions of EdgeX.
- The service should not rely on new or variant APIs.
- The service's name should should follow the usual form, ie edgex-{device-class}-{language} and not conflict with any other adopted service.
- Values must not be hardcoded where they might reasonably be configurable.
- If the devices may connect via a secure mode, credentials must be obtained via the Secrets API rather than the service configuration
- A top-level README should be present and contain information on
    - The types of device supported

    - Host device requirements, especially if advanced features or operating modes are needed.
    - Run-time dependencies
    - Protocol feature support and roadmap
    - Known limitations
    - Information on asynchronous readings, if these are generated
    - Whether dynamic discovery is supported, and if so what limitations or special requirements apply
    - Build instructions, including build-time dependencies
    - Usage information (command-line options)
- The following items must also be documented:
    - Supported configuration options (including whether or not the service must be restarted for changes to take effect)
    - Supported ProtocolProperties schemes.
    - Supported Device Attributes.
- As part of the above, example Device Profiles and illustrative Device definition files should be included.
- It must be possible to run the device service against simulated hardware. Documentation illustrating how to do so is also required. The review group should attempt to replicate the scenario described.
- Real hardware against which the service has been tested should also be documented.
- Container packaging must use full confinement (i.e. no use of docker --privileged or snap --devmode). Where access to hardware is required, care must be taken to make exceptions for the specific required hardware and/or system resources, and the means to allow these exceptions should be documented.
- A default listening port will be assigned to the new service. Sample configurations supplied with the service should specify this port.
- The service should comply with the general EdgeX requirements as given in the Contributor's Guide.

Further actions following adoption

- EdgeX documentation is updated to include the new service.
- Any references to the -holding repository (eg in Makefiles, go.mod etc) need to be updated.
- Required devops infrastructure is put into place.
- Release artifacts for the first version should be generated and published.
- Device WG wiki pages to be updated.